International Journal of Computer Engineering & Technology (IJCET)
Volume 7, Issue 2, March-April 2016, pp. 01-14, Article ID: IJCET_07_02_001
Available online at
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=2
Journal Impact Factor (2016): 9.3590 (Calculated by GISI) www.jifactor.com
ISSN Print: 0976-6367 and ISSN Online: 0976–6375

TEMPERATURE CONTROL AND DATA ACQUISITION METHOD FOR FACTORY USING LABVIEW

Yusuf Ayuba
Department of Computer Engineering,
Faulty of Engineering, University of Maiduguri,
Borno State, Nigeria

ABSTRACT

The Aim of this paper is to present both automatic and manual temperature control system for modern data acquisition processes. In developing the work, consideration of steps' sequence was done; stating at the beginning point which is reading of temperature input using LM35 temperature sensor. Then convert that input analog into digital using Arduino microcontroller. After conversion, we do the appropriate task, for example, if the temperature reading is greater than factory temperature, the cooling unit will ON automatically and give us a mean to regulate the cooling unit speed. The software (LabView) that was used in this design has capability of manual regulation controls and responding in real time to takes the next step further to run, store and show the result schemes based on graphical temperature chart using the PC. The PC also stored the readings for the data acquisition result that is being exported into Microsoft Excel.

Key word: LabView, Storage, Automatic / Manual control, Hardware and Software, Data Acquisition, Speed Control etc.

Cite this Article: Yusuf Ayuba. Temperature Control and Data Acquisition Method For Factory Using Labview. *International Journal of Computer Engineering and Technology*, **7**(2), 2016, pp. 01-14.
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=1

1. INTRODUTION

The principle of division of labour is visible in using technology to assist man in the disposition of tasks with little effort using control systems. Ordinary control system measures the controlled variable, compares that measurement with the set point or reference, and if there is a difference between the two, changes its output signal to the manipulated variable in order to eliminate the error [1]. This however gives enormous advantages as great amount of stress and difficulties which can be monotonous are

overcome using a relevant applied system. Therefore, from the principle of the control systems, this focused on the domain of Temperature Control system.

Ideal factory temperature suitable for both production and storage of goods, is usually adversely affected by too high or too low temperature which are cumbersome to regulate and control using conventional methods, hence the need for computerize and efficient system development to achieve the task is will be needed. Getting the precision temperature measurement daily and taking note of the environmental and location rise or fluctuation in temperature is also a challenge for many factories (e.g. pharmaceutical). However, the accuracy of these measurements will be meaningless unless the equipment and sensors are used correctly. Acquired data to be processed and stored for maintenance purpose and analysis in the future is also a crucial issue. The manual methods used for data acquired records are most likely to be prompt to errors due to human miscalculation and inability to write down the correct figures of the records obtain. Lab view „G" program allows users to create a graphical program which process the data obtained from the sensor (e.g., temperature sensor, pressure sensor, etc) [2], [7].

2. TEMPERATURE CONTROL AND DATA ACQUISITION

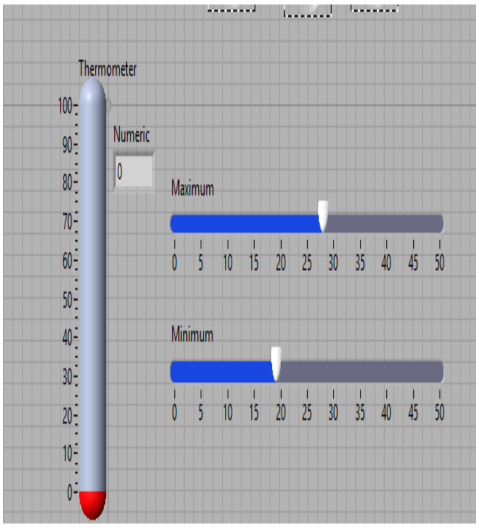
2.1 Temperature control

Accurate control of temperature is essential nearly in all chemical processes [4], but also varies in some applications. An accuracy of around 5-10ºC may be acceptable. There are some industrial applications which require better than ±1ºC accuracy. Temperature control is important for separation and reaction processes, and temperature must be maintained within limits to ensure safe and reliable operation of process equipment [3]. Various temperature scales have been proposed since time; in the centigrade, or Celsius, scale, devised by the Swedish astronomer Anders Celsius and used in most of the world, the freezing point is 0°, the boiling point is 100°. Celsius and Fahrenheit temperatures can be interconvert as follows:

$C = (F - 32) \times 100/180$; $F = (C \times 180/100) + 32$.

Celsius and Kelvin can be interconverted as follows:

$C = (K - 273.15)$; $K = (C + 273.15)$.



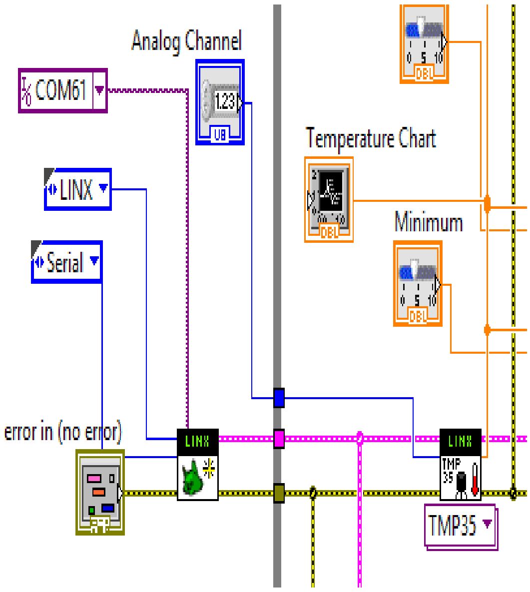


Figure 2.1 Front and Back Panel of LabView temperature controller

2.2. Data Acquisition

Data Acquisition (DAQ) is the principle of conversion using I/O signal interface from one device into another (e.g. Analog to Digital) using the appropriate channel of conversion. Data acquisition systems have evolved over time from electromechanical recorders containing typically from one to four channels to all-electronic systems capable of measuring hundreds of variables simultaneously [8]. Most DAQ can be categorize as either external or plug – in board configuration. As the name imply, Ethernet and USB are external system, while PCI and PCI Express are considered internal "DAQ Glossary". LabView VI uses USB cable to connect the port for data acquisition board (Ardouino), while another side of the cable connects to the port of PC which shows the COM Port communication number when tested to the labview connection port block, which in this case is tested to be COM65 as shown in figure 2.2.

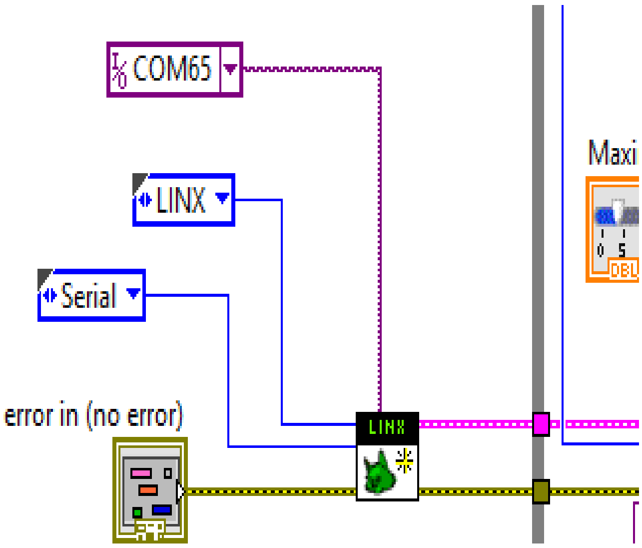


Figure 2.2 LabView DAQ communication ports.

All PC-based data acquisition systems will record extremely accurate, repeatable, reliable, and error-free data provided they are connected and operated according to the manufacturer's recommended practices [8]. Configuring write to measurement of file gives the PC directory to channel for the interface transfer of the files for DAQ from the microcontroller (Arduino). Through the Functions Palette (Express > Output > Write to Measurement File) place on the block diagram. A dialog box will appear. In the dialog box, pick a convenient directory for the output file. Choose File Format = Microsoft (.xlsx), X Value Columns = One column only, Delimiter = Comma, and If a file already exists = Overwrite file.

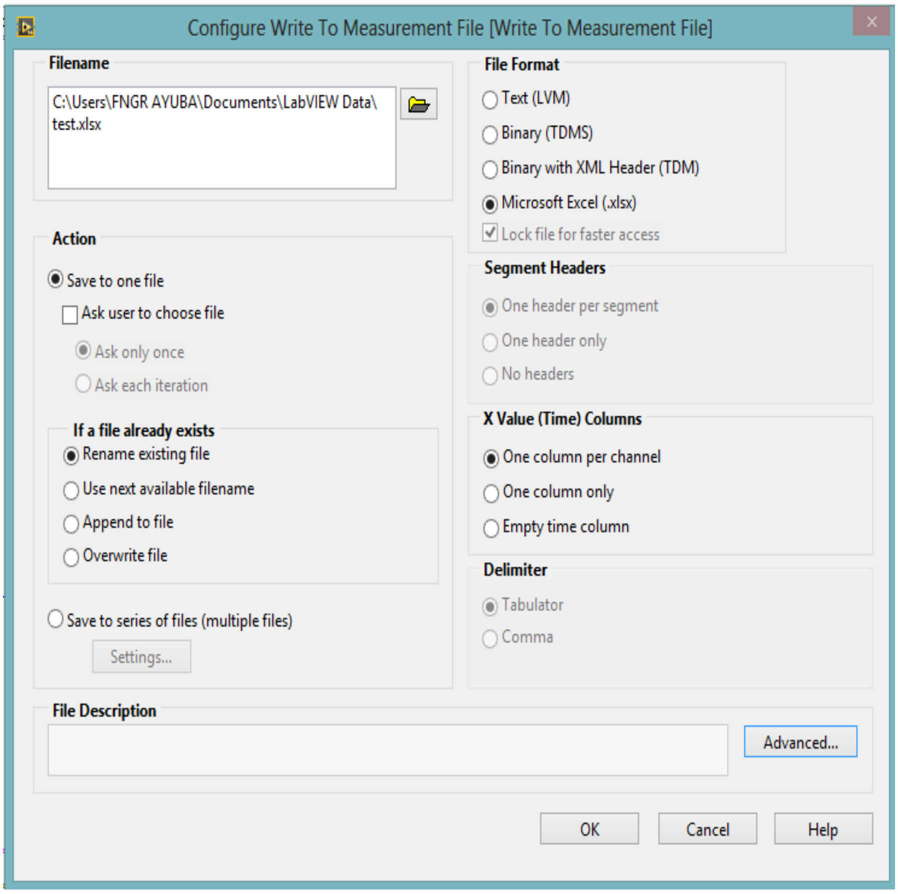


Figure 2.2 Write to measurement file functional palette.

3. HARDWARE AND SOFTWARE PROCEDURE

3.1. HARDWARE DESISGN

The under listed component where used for the implementation of the design. The choice of these components was based on the needs and availability.

- Arduino Uno
- Resistors
- Temperature sensor (LM35)
- RGB LED
- Cooling fan (5V)
- Buzzer
- Switch

3.1.1. The Temperature Sensing Unit (Lm35-D)

The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in degree Celsius); the sensor also has a high precision and wide linear range. Therefore, comparing it with the Kelvin standard linear temperature sensor.LM35 has more advantages, for the most important part, it does not need external adjustment or trimming, the sensor allows the temperature between -55~150^0 C, could provides the common room temperature precision of $\pm$ 4^0 C. The LM35 has an output of 10mV/°F with a typical nonlinearity of only ±0.35°F over a −50 to +300°F temperature range, and is accurate to within ±0.4°F typically at room temperature (77°F) [5]. This sensor output voltage is linearly proportional to the Celsius. It has 3 pins, pin1 is connects to Vcc for power supply from the Ardiuno microcontroller 5v, pin2 is connects to the analog pin A0 of the microcontroller for the digital transition, while pin3 is connects to the ground.

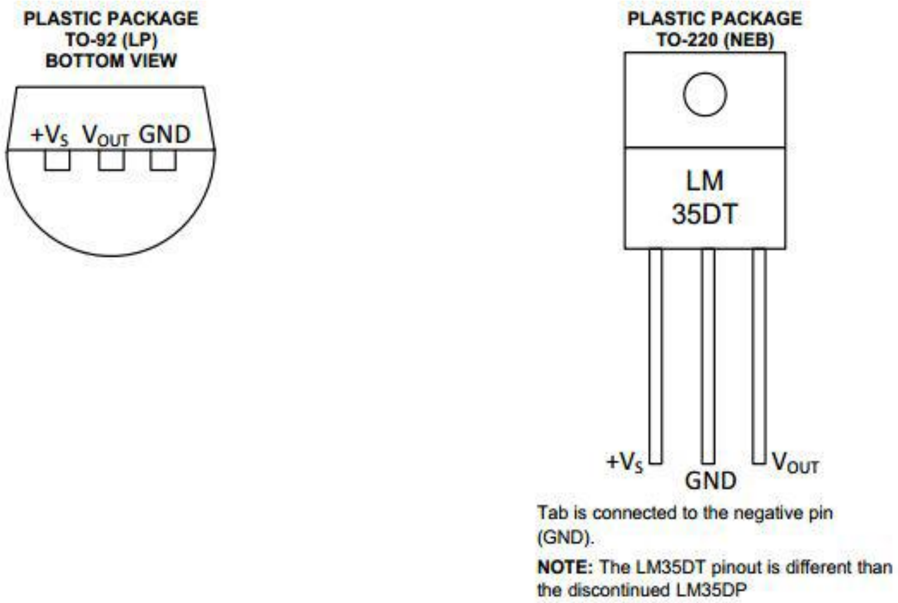


Figure 3.1 LM35 temperature sensors

3.1.2. Led Indicator Unit

The LED (Light Emitting Diode) indicator unit comprises of three LEDs. The first one is a green LED and is connected directly to the end of the resistors pin, which is power and triggers from arduino DO9 pin; hence indicating when the system is moderate. The second is yellow and is connect DO 10 pin and control indication when temperature is Low. The third LED is red, its turn digital High (DO 13) simultaneously with the cooling system unit and the sounding alarm as the reading exceeds certain preset value. Figure 3.3 shows the RGY LED hardware implementation.

3.1.3. Cooling Units

A 5V DC fan was used to demonstrate the output of the control system. After it receiving input signal from the arduino microcontroller (DO 8), it give the output to the cooling fan. The fan terminal is wired and connected to one of the LED logic pin design in the labview, which both start automatically at rise in temperature greater than the factory's temperature.

3.1.4. The Buzzer Alarm

The buzzer is used to convert electrical signals into sound just like a speaker. It is generally used to generate electronic sounds or beeps from a microcontroller by connecting its positive terminal directly to a pin of the microcontroller. In the case of this work, the buzzer two wire are connected to pin 13 (red wire) and pin GND (black wire) of the Arduino microcontroller, digital output which goes HIGH whenever the temperature readings goes high beyond a preset value.

Figure 3.2 A Buzzer

HARDWARE PROCESS SETUP

The process is a cascade control setup shown in figure 3.3

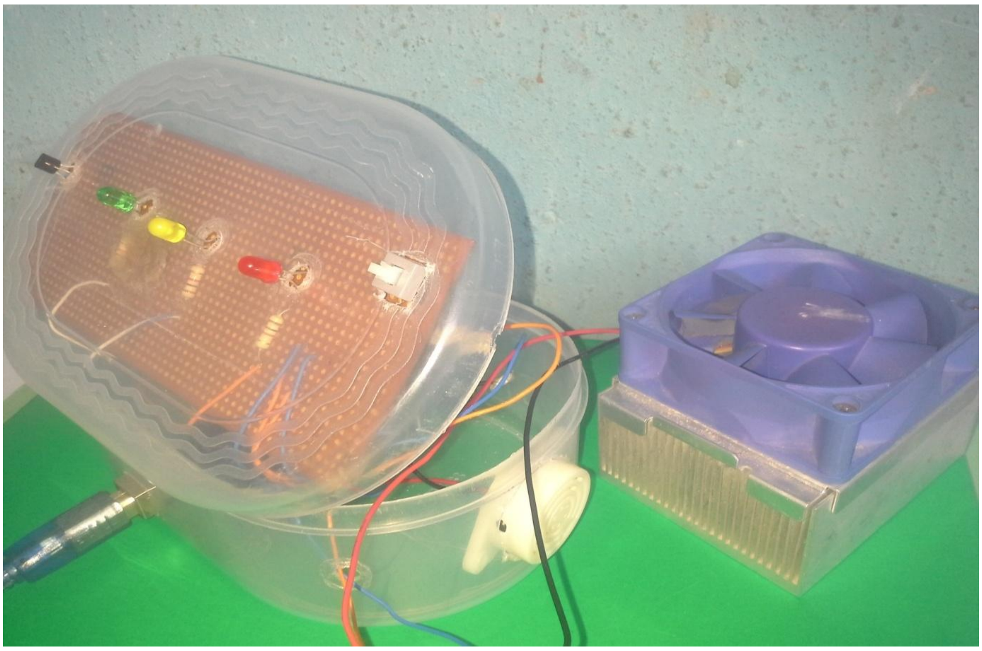


Figure 3.3 Hardware setup

3.2. SOFTWARE DESISGN

The first step in this section is the programming of the LabView Virtual Instrument (VI) to sample the temperature from the construction circuit divider and convert it

into a temperature reading in real time. But this process should be distinguished from real time simulation designs [6]. LIFA LabView interface For Arduino (or Linx) is the software that helps us interface LabView and the Physical Arduino Hardware design. To achieve this, two steps were included, first creating the conversion section then creating the data acquisitions setting. The Microsoft Excel serve as where the data obtained from the temperature environment is being written to. A VI consists of a front panel, block diagram, and an icon that represents the program. The front panel is used to display controls and indicators for the user, and the block diagram contains the code for the VI. The icon, which is a visual representation of the VI, has connectors for program inputs and outputs [7].

3.2.1. Front Panel Design

The front panel allows us to control and monitor the process. It consists of software controls and indicators that resemble the physical controls of the hardware construction such; as LEDs, sliders, buttons, speed controller and graphic charts. Figure 3.4 is a screenshot of the front panel.

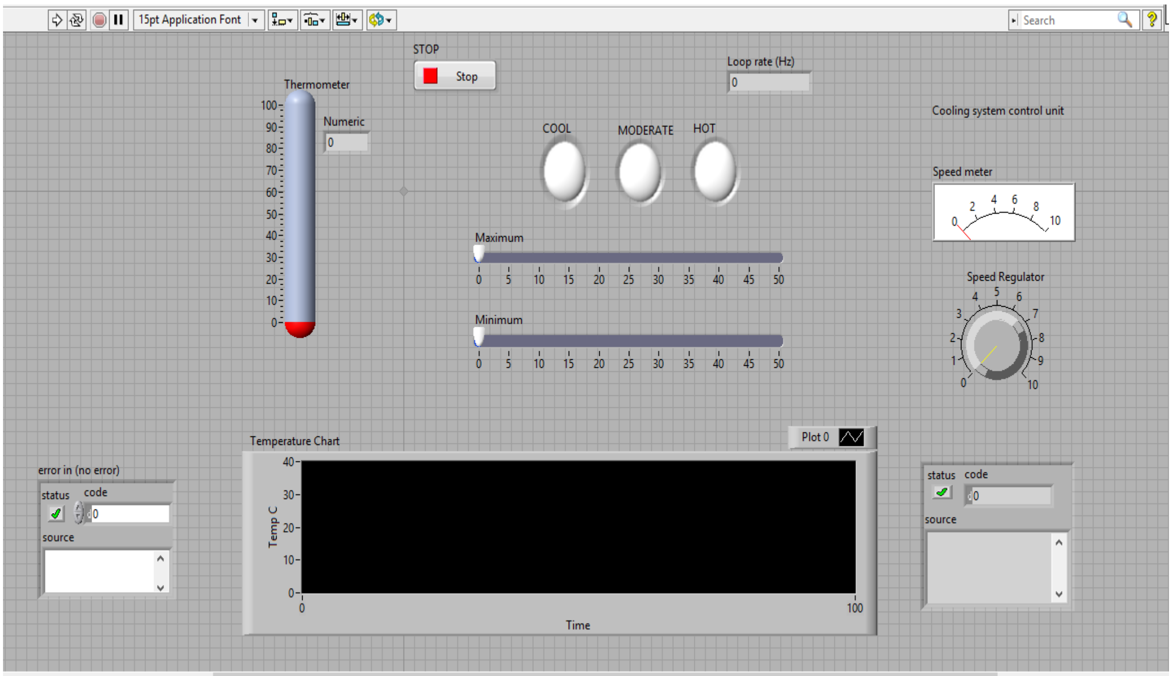


Figure 3.4 Front Panel

3.2.1. Back Panel Design

The Back Panel is where to constructs the Block Diagram, which is the graphical program that shows the data flow of the temperature control operation. Unlike a high level language program, like the C language [7] where instructions are executed in the order that they are written, the execution of a LabView VI depends solely upon the flow of data: a particular object inside the Block Diagram execute in by the available component that we made present at the front panel. Each node was connected properly in this part to avoid errors. Figure 3.5 shows the details of the Block Diagram which describe the operation.

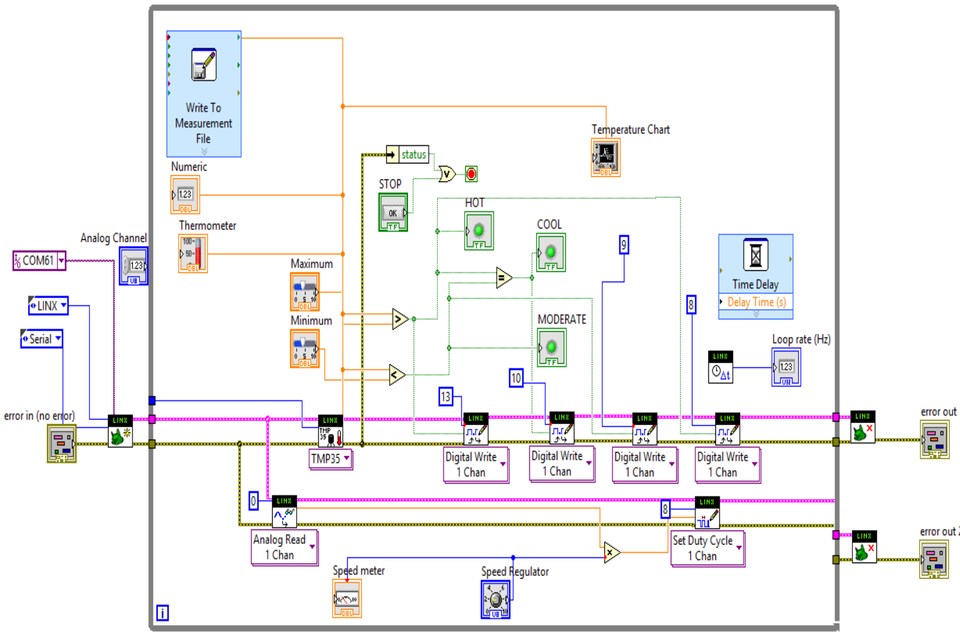


Figure 3.5 Back Panel

Finally checking how the program runs in the temperature control map by building and running the test icon on left top side. If there is a bridge of program flow, the back panel also give a sign in other to make the appropriate error debugging connection.

3.2.3. HOW TO INTERFACE ARDUINO WITH LABVIEW

LINX device driver must be installed in order to interface the Ardoinu microcontroller port. LabView communicate with the Ardoinu microcontroller by using its LIFA or LINX. The value obtained from the temperature sensor is displayed in LabView module. LabView module initializes the communication by accessing the temperature sensor of the embedded program using the install drive driver. Bellow is steps to follow for the driver installation process [9];

- **Step 1:** Download LabVIEW Interface For Arduino (LIFA)
- **Step 2:** Don’t install LIFA directly. If done so an error like “VIPM could not continue” will occur.
- **Step 3:** Open LabVIEW go to Tools > options > VI server
- **Step 4:** After clicking on VI Server scroll down to check Machine Access. Then add machine access list manually. Make sure that you added “10.2.11.55.13”, “localhost”, “*” in the Machine access list, then click on ok.
- **Step 5:** Now open downloaded VI Package Manager.
- **Step 6:** Search for LabVIEW Interface for Arduino and double click on it to install it.
- **Step 7:** Click on continue and after installation is, completed click on finish.

4. RESULTS AND DISCUSSION

After successful implementation and test run, the following figures and tables show the temperature variable range and controller output plotted with respect to each time chart. Figure 4.1: The output result shows when the factories environment temperature is fluctuating at 29.78^{0}C, while the temperature range requirement is between10~35^{0}C, therefore the cooling system unit does not trigger ON.

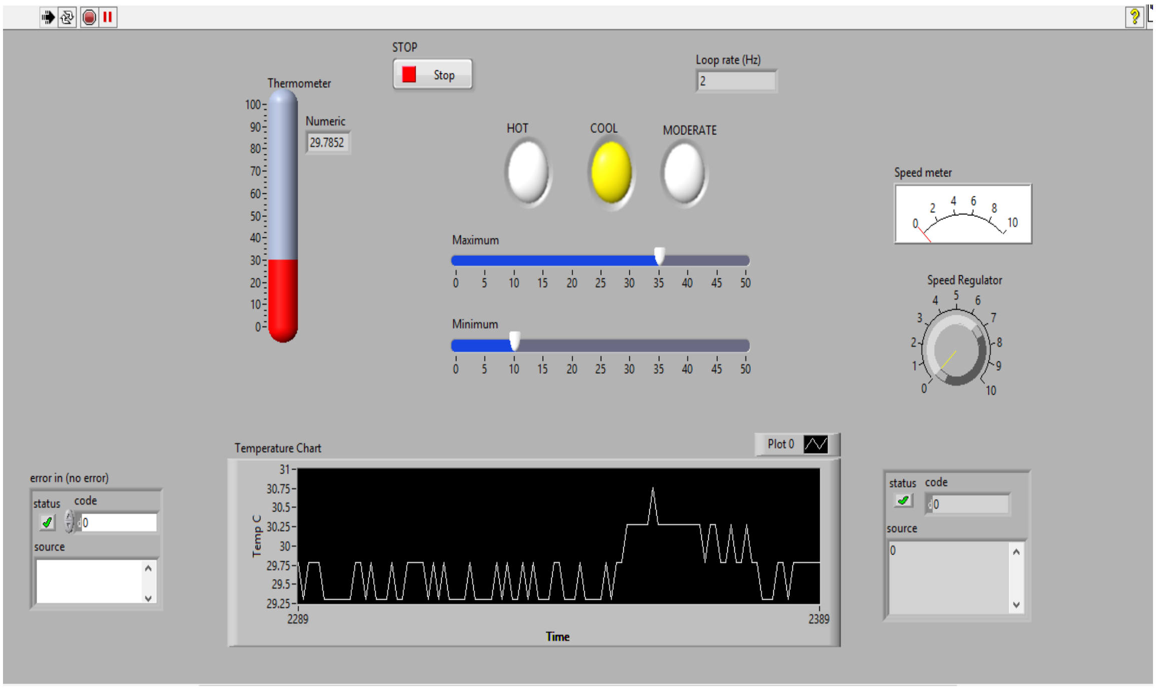


Figure 4.1

Figure 4.2: The output result shows when the factories environment temperature is still fluctuating at 29.78^{0}C, while the temperature range requirement from the Max and Min slider is 40^{0}C. This example shows that in some factories “for a reliable factory Automation, operating temperature range from 0 to 40^{0}C” [10]. Therefore the cooling system unit still remained OFF and does not trigger.

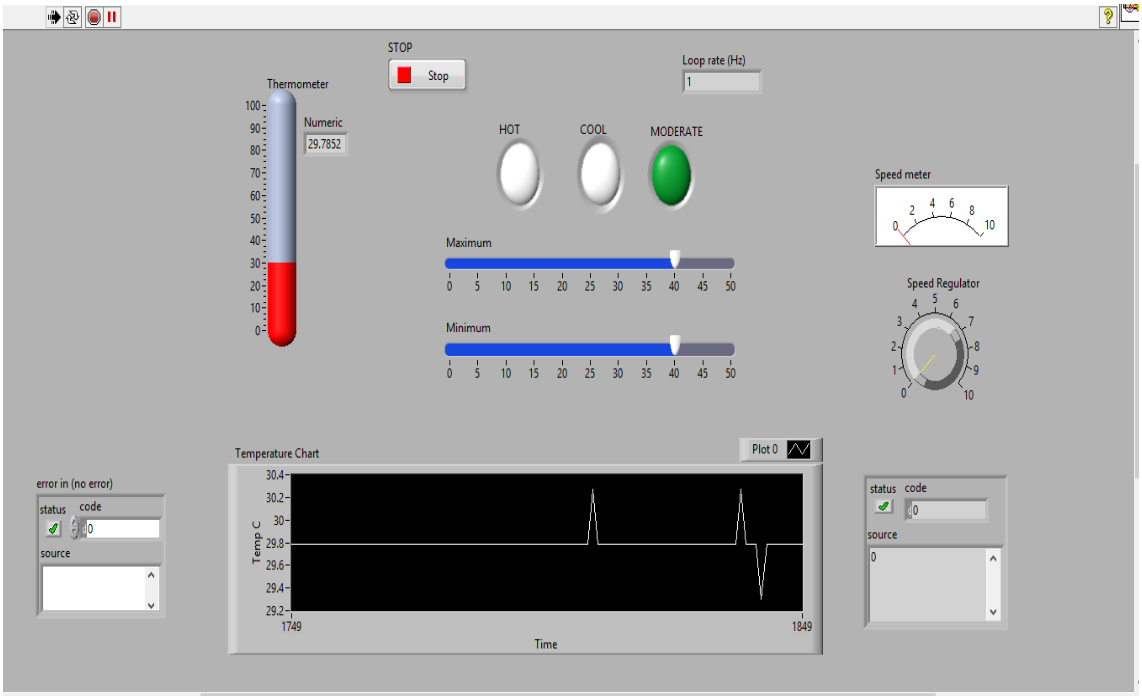


Figure 4.2

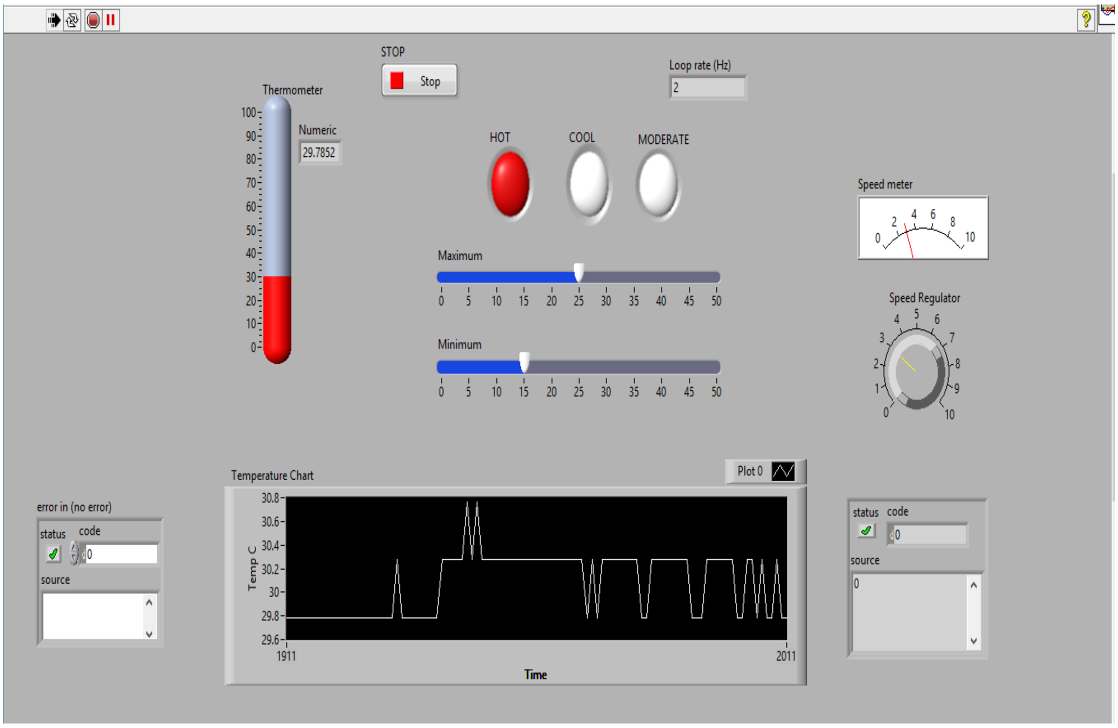


Figure 4.3

Figure 4.3: The output result shows when the factories environment temperature is still fluctuating at 29.78^{0}C, while the temperature range requirement from the Max and Min slider in this case is between 15~25^{0}C. Therefore since environment temperature is greater than the required temperature, this triggers the cooling system unit ON and shows the speed level on the speed meter which can be regulated using the virtual regulator.

The following figures show the Virtual regulation with respect to speed generated.

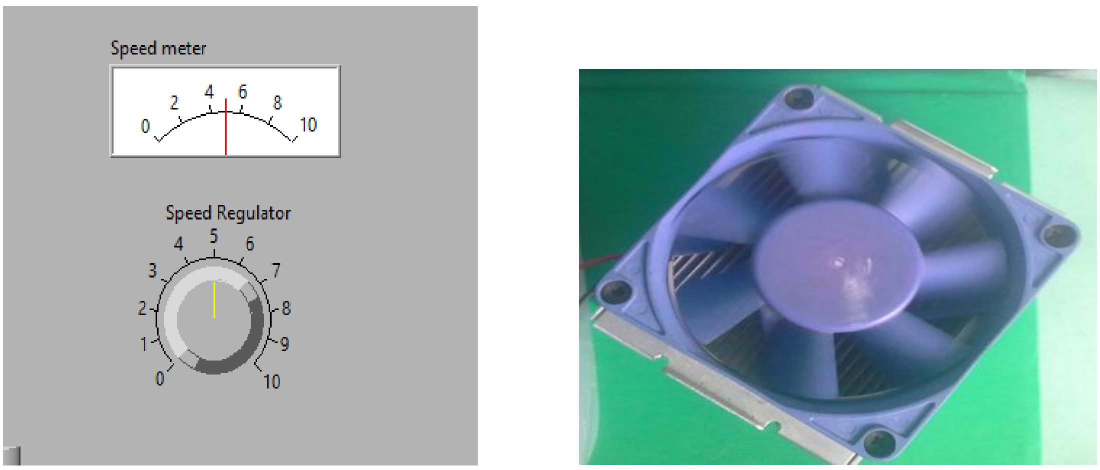
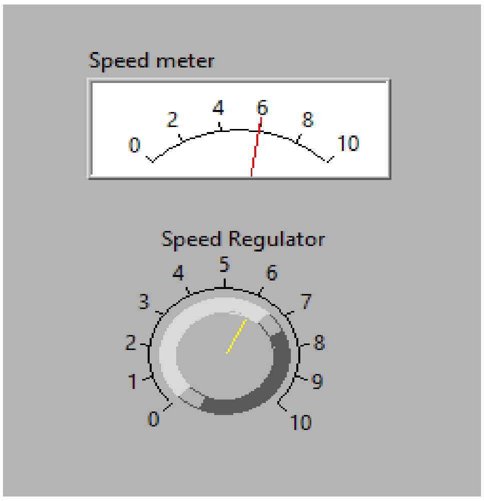


Figure 4.4 Regulation at 5



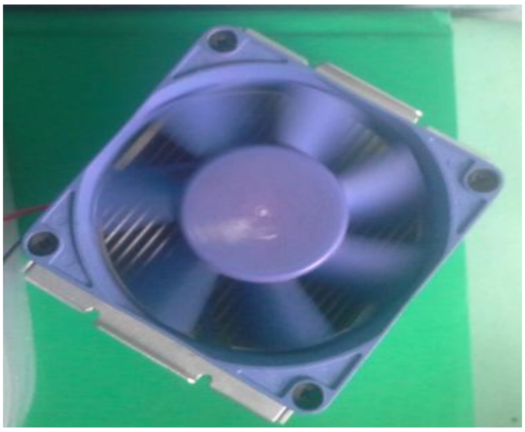


Figure 4.5 Regulation at 6

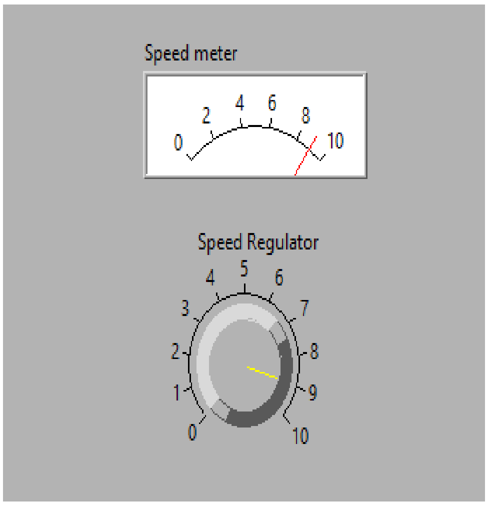


Figure 4.6 Regulation at 9

Table 1 Data obtain during first test

Date	**Time**	**Temp O C**
11/07/201515	15:15:24.000	29.04297
11/07/201515	15:15:24.437	30.01953
11/07/201515	15:15:24.642	29.53125
11/07/201515	15:15:24.942	29.53125
11/07/201515	15:15:25.220	30.01953
11/07/201515	15:15:25.516	31.87134
11/07/201515	15:15:26.072	32.00135

Table 2 Data obtain during second test

Date	Time	Temp ^{O} C
16/07/2015	22:07:56.989	31.73828
16/07/2015	22:07:59.456	32.64828
16/07/2015	22:08:32.489	33.11125
16/07/2015	22:08:49.438	32.53125
16/07/2015	22:08:56.237	33.01953
16/07/2015	22:09:23.789	33.87134
16/07/2015	22:09:43.403	34.00135

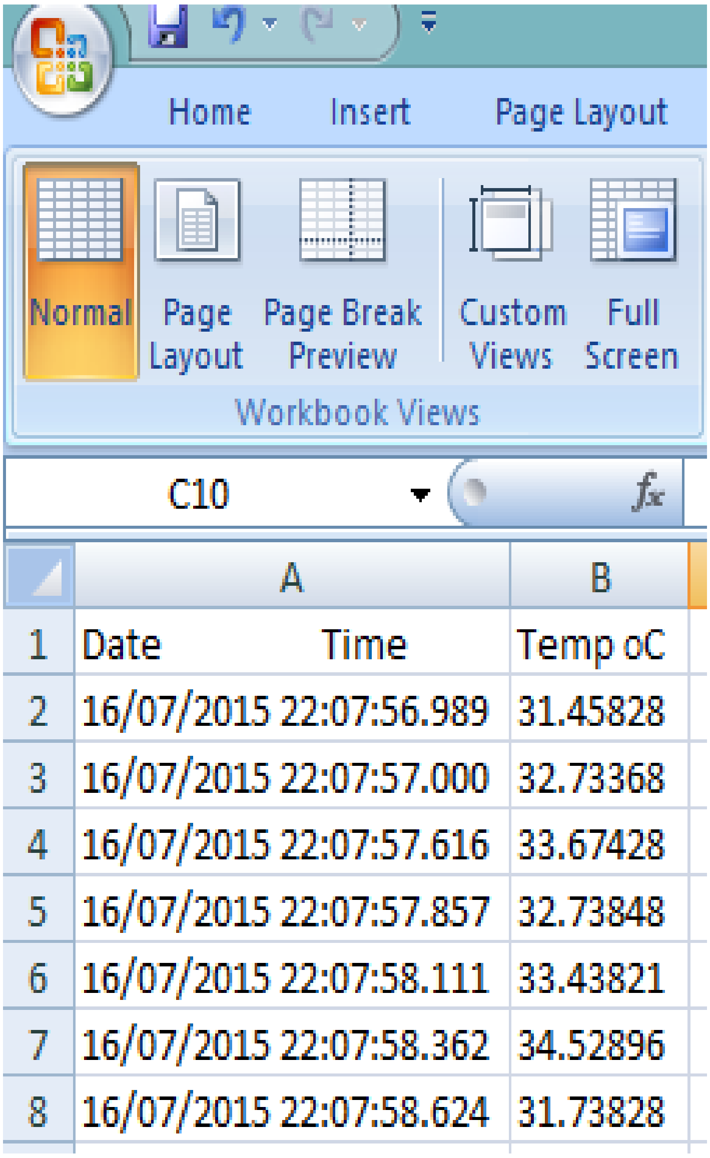


Figure 4.6 Data Storage in Excel

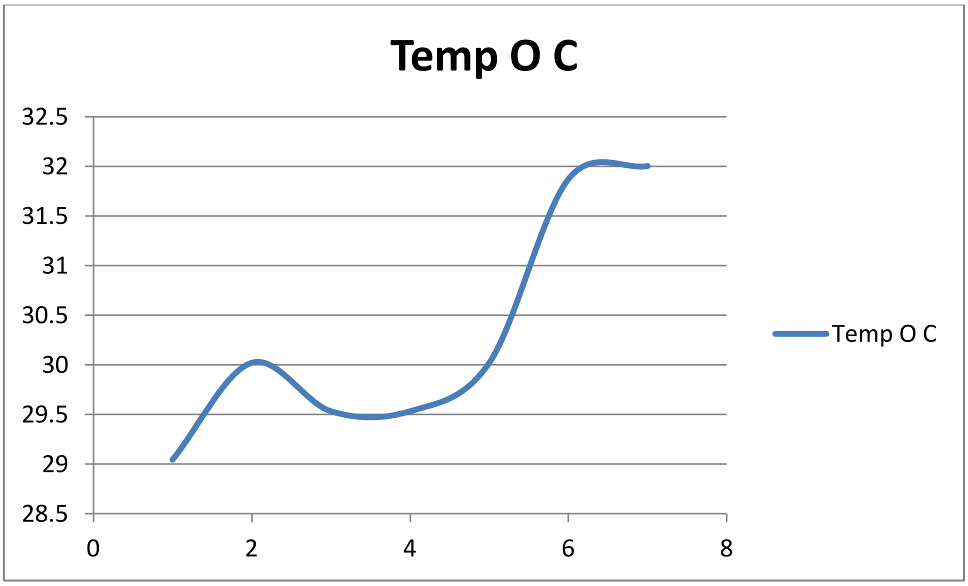


Figure 4.7 Graphic date Exported from Excel

5. CONCLUSION

The main aim of the design which is to both automatic and manually read and regulate the temperature of factory environment and to store and display it on LabVIEW was however been achieved. Also all types of controllers are designed in the LabVIEW. There may be other softwares used for designing control system but LabVIEW is the simplest of them all. Is because it uses the drag and drop principle, it doesn't need any code to run the software since it follows graphical coding E.g. for a while loop we simply make a box inside which the contents of the statement are taken from. I also recommend future design to incorporate other features such as air-condition and weather station controller and advance sensing unit, to be tried and tested using this work in other to substitute the manual and analog form of data acquisition. While future work still remains, this work serves as a powerful learning experience.

ACKNOWLEDGEMENT

My profound gratitude and appreciation goes to my supervisor and the Head of the Department of Computer Engineering in person of Engr. Dr. Christopher U. Ngene for sparing his valuable time in going through this work and paper which he give an evinced keen interest in solving my problems and bringing it to its final stage. Indeed He is a mentor, good listener, straight forward, and a strict disciplinarian, I consider myself fortunate to have worked under his supervision, because without him this paper would not be possible for publication. My unreserved appreciation and sincerely grateful to my lecturer Mr. Peter Y. Dibal, for his tutoring, counselling and advice that he bestowed upon me. And moreover for believing in me, that even nobody student who come from low local area like me can contribute his quarter in to the world of Academe. I will always take a bow before saying thank you sir.

REFERENCES

[1] P Srinivas, P Durga Prasada Rao, K Vijaya Lakshmi (2012), "Modelling And Simulation Of Complex Control Systems Using Labview" International Journal of Control Theory and Computer Modelling (IJCTCM) Vol.2, No.4, July 2012

[2] Rajesh, S. K. (2013), ‘‘Industrial Temperature Monitoring and Control System through Ethernet LAN ’’International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 2 Issue 6 Page No. 1988-1991

[3] Janani Rajaraman, K.Saraswathi "A Labview Based Monitoring and Controlling of Various Process Variables" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering *(An ISO 3297: 2007 Certified Organization)* Vol. 3, Issue 10, October 2014

[4] George Stephanopoulos, "Chemical Process Control – An Introduction to Theory and Practice", ISBN – 978-81-203-0665-3

[5] AN-460 LM34/LM35 Precision Monolithic Temperature Sensors, Application Report *SNOA748C–October 1986–Revised May 2013*

[6] Shyam, R., Nair, V., and Venkatesh, T. (2012) "Design of a LabVIEW Based Real Time Simulator for ABS Tester", International Journal of Scientific and Research Publications, Volume 2, Issue 5, May 2012.

[7] Rick Bitter, Taqi Mohiuddin, Matt Nawrocki "LabVIEW Advanced Programming Techniques" second edition ISBN-13: 978-0-8493-3325-5 (Hardcover) p22, 1.

[8] Data Acquisition Handbook, A Reference for DAQ and Analog & Digital Signal Conditioning Third edition, Published 2004-2012 in the United States of America by Measurement Computing Corporation p 1-2.

[9] S.Hema Chandra,Dr.R.V.S.Satyanarayana. Temperature Control of Transformers Using Soft Computing Techniques. *International Journal of Computer Engineering and Technology*, **3**(2), 2012, pp. 133-137.

[10] Joshi M.P, More V.A.. Real Time Cost Effective Temperature Controller. *International Journal of Electronics and communication Engineering and Technology*, **3**(2), 2012, pp. 271 - 277.

[11] How to Interface Arduino with Labview, [online]. Available from: http://myclassbook.org/labview-interfae-for-arduino [Accessed 18th May, 2015]

[12] Factory operating in high temperature " Increased Holiday Demand Highlights Need for Reliable factory Automation", Axiomtek Co., Ltd 2014, p 2. Www. Axiomtek.com